Algorithm

Annie Luciani

Algorithm

Annie Luciani [ACROE&INPG]

Contributors: Damien Courousse [ACROE&INPG]

When mediated by computer, computation constraints and rules are introduced within the relation between humans and environment. This relation is no longer direct, in the sense that it no more depends only on human properties and external mechano-optical world, but it depends also of computation properties.

Computation constraints can be due to the hardware components such as when mechanico-optical objects are equipped by sensors and actuators. They can also be due to the formal aspects of the computation itself: a program that corresponds to a formal mathematical model able to be implemented on a specific computation automaton, called computer.

This specific mathematical model was called algorithm, from the name of the Arab mathematician Al-Huwarizmi. By his works, Al-Huwarizmi led to define a part of Mathematics, called nowadays the science of algorithms, or algorithmics, which aim is to l*et* a mathematical problem, for which mathematical demonstrations have proved that a solution exists, find the solution under defined constraints.

Constraints can be various: number of steps, fixed time, memory occupancy, repartition of calculus over several computers, etc, leading to researches in computational optimization, parallel computation, real time computational techniques, etc.

The transformation of a mathematical expression of a problem - usually called model in experimental sciences as physics, biology, etc. - into an algorithm and, further, into a real computer implementation and real computation, introduces a cascade of causalities between the successive computed data, that further introduces non trivial transformations in the human-world interaction. Designers and users of virtual realities systems, human-computer interfaces and above all enactive interfaces, have to be aware of:

- First, the fact that algorithmic representation is very different than some other mathematical representations. The design of the algorithms consists in transforming the mathematical expressions of the model in a set of sequential mathematical computable actions.
- Second, the fact that to be computed by a computer, the algorithm needs to be transformed into a computer program that allows the mathematical actions to be automatically performed by the components of the computer (addition, multiplication, transfer, test, etc…). Designing a computer program consists in translating the algorithm into a programming language, so that the computer can perform the actions described by the algorithm.

A couple of examples can help clarifying the impact of these transformations in the domain of enactive interfaces.

First, let us consider a system that is not an algorithm. The Watt's centrifugal governor, considered as basic first cybernetics system [→ Dynamic systems] [→ Cognition, dynamic systems approach] is a physical system that implements physically a control function [Clark & Toribio, 1994]. It is not an algorithm. To be computable, this function requires to be expressed in a formal set of sequential operations.

A more emblematic example is given by the case of a physically-based model used to create a physical virtual object, that is felt by humans through a force feedback device.

Let us consider the simple physical system made of two masses in elastic interaction. A usual physical mathematical representation is based on equations that link two expressions through an equal sign:

Mass 1 (and 2): $F= M*a$
where a is the acceleration of each masses;
Elastic interaction: $F= K*d$
where d is the distance between masses.

The equal sign implies that there is no causality between force and position; one cannot say that force causes position, nor the opposite.

Unfortunately, the equal sign does not exist in algorithmic representation. These equations have to be transformed into input-output expressions, each expression being a sequential action. For example:

First: *having a, compute F*
Then: *having F, compute d*
Then: *having d, compute a …*

Hence, the transformation into an algorithm necessarily translates a physical non-causal representation into a causal representation. In this precise case of computation of physical rules, the physical principle of non-causality (for example between force and position) that is fundamental in physics, is impossible to satisfy.

Moreover, when this virtual physical object is linked to a force feedback, the causality cascade is worsened again by a temporal causality (the incompressible latencies) between data (positions for ex.) provided to the program by the external device and data (forces for ex.) sent back to the external device [→ Channel, afferent / efferent] [→ Force].

Indeed, when computed by a machine, each step of the algorithm requires a certain time, not equal to zero. A temporal delay – which was not taken into account in the initial mathematical model - is introduced between the real inputs and outputs of the calculus. This distorts further the correspondence of the computed results with the real observed phenomena that was initially represented by the mathematical equations based on the equal sign.

There exist other transformations, such as numerical approximations, that are usually known by designers and users in virtual realities and human – computer interfaces. However, the transformations into an algorithm, which is studied by disciplines as real time simulation and robotics, are often less known in virtual realities and human – computer interfaces. In the domain of enactive interfaces, being aware of the effects caused by an algorithmic representation and implementation is of a central and critical importance.

References

[Clark & Toribio, 1994] Clark Andy & Toribio Josepfa. "Doing without representating?". Synthese 101:1994 401-431. Kluwer Academic Publishers. 1994.

Related items

Channel, afferent / efferent
Dynamic systems
Force
Simulation
Stability
Turing machine

Related documents

EI_Enaction&ComputerTechnologies_Luciani.pdf
EI_InputOutputParadigm_Luciani.pdf